FOSS Static Analysis Tools for Embedded Systems and How to Use Them

Jan-Simon Möller, The Linux Foundation

jsmoeller@linuxfoundation.org

ELCE 2020

THE LINUX FOUNDATION

Intro

Dipl.-Ing.
Jan-Simon Möller

dl9pf on freenode
dl9pf@gmx.de

AGL Release Manager
jsmoeller@linuxfoundation.org
Yocto Project Board Member

Topics

- Intro & Motivation
- kernel vs. userspace
- local tools
- meta-sca - a collection of tools
- meta-codescanner - clang-sa/clang-tidy integrated
- Summary & lookout
- Q/A

Motivation

- Static Analysis is a method to analyse a program that is performed without actually executing programs.
- Static Analysis becomes an increasingly important topic when the project involves Functional Safety aspects. This is the case in Automotive and in Automation as well.

Motivation

- In the case of AGL or ELISA, we have to fulfill and document requirements on code quality for own and reused OSS code.
- The goal is to show ways how to ensure this using open source tools available.
- I will introduce basics but focus on what can be integrated with OpenEmbedded / the Yocto Project builds.

kernel & userspace

The Linux kernel is a very large and special codebase.

Currently it contains more than 20 million lines of code. This is very demanding on the tooling used. Thus there are specialized tools around the kernel:

- scripts/checkpatch.pl (string matching, basics&style, good for new submissions)
- gcc / clang static analyser
- sparse `make C=1 CHECK="/usr/bin/sparse"`
- smatch `make C=1 CHECK="smatch -p=kernel"`
- coccinelle `make C=1 CHECK="scripts/coccicheck"`

  (can patch code or just warn on patterns)

Proprietary: e.g. Coverity, CodeSonar, SonarQube

kernel & userspace

For userspace there are a large number of tools available. A selection for C/C++ is below:

- gcc
- clang
- cppcheck
- flawfinder
- rats
- splint

local tools

During development you can easily use these directly within your source tree:

- gcc (since gcc 10)
  - gcc -fanalyzer
- clang
  - e.g. scan-build make
- cppcheck

gcc -fanalyzer enables:

-Wanalyzer-double-fclose
-Wanalyzer-double-free
-Wanalyzer-exposure-through-output-file
-Wanalyzer-file-leak
-Wanalyzer-free-of-non-heap
-Wanalyzer-malloc-leak
-Wanalyzer-possible-null-argument
-Wanalyzer-possible-null-dereference
-Wanalyzer-null-argument
-Wanalyzer-null-dereference
-Wanalyzer-stale-setjmp-buffer
-Wanalyzer-tainted-array-index
-Wanalyzer-unsafe-call-within-signal-handler
-Wanalyzer-use-after-free
-Wanalyzer-use-of-pointer-in-stale-stack-frame

gcc

```
> gcc -Werror -fanalyzer nullpointer.c
nullpointer.c: In function ‘main’:
nullpointer.c:7:5: error: dereference of NULL ‘pointer’ [CWE-690]
[-Werror=analyzer-null-dereference]
      7 | int value = *pointer; /* Dereferencing happens here */
        |     ^~~~~
  ‘main’: events 1-2
        |
        |     6 | int * pointer = NULL;
        |       |     ^~~~~~~
        |       |     |
        |       |     (1) ‘pointer’ is NULL
        |     7 | int value = *pointer; /* Dereferencing happens here */
        |       |     ~~~~~
        |       |     |
        |       |     (2) dereference of NULL ‘pointer’
        |
cc1: all warnings being treated as errors
```

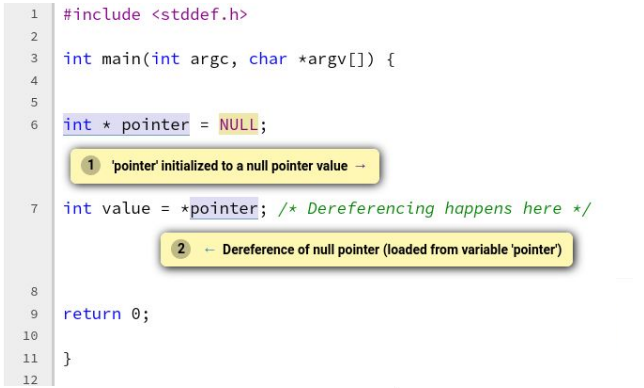
clang (clang-tidy)

```
> clang-tidy nullpointer.c
Running without flags.
2 warnings generated.

nullpointer.c:7:5: warning: Value stored to 'value' during its initialization is never read [clang-analyzer-deadcode.DeadStores]
int value = *pointer; /* Dereferencing happens here */
    ^
nullpointer.c:7:5: note: Value stored to 'value' during its initialization is never read

nullpointer.c:7:13: warning: Dereference of null pointer (loaded from variable 'pointer') [clang-analyzer-core.NullDereference]
int value = *pointer; /* Dereferencing happens here */
            ^


nullpointer.c:6:1: note: 'pointer' initialized to a null pointer value
int * pointer = NULL;
^

nullpointer.c:7:13: note: Dereference of null pointer (loaded from variable 'pointer')
int value = *pointer; /* Dereferencing happens here */
            ^
```

clang (scan-build)

```
> scan-build make

scan-build: Using '/usr/bin/clang-10.0.1' for static analysis
/usr/bin/ccc-analyzer  -c nullpointer.c -o nullpointer

nullpointer.c:7:5: warning: Value stored to 'value' during its initialization is never read
int value = *pointer; /* Dereferencing happens here */
    ^~~~~   ~~~~~~~~
nullpointer.c:7:13: warning: Dereference of null pointer (loaded from variable 'pointer')
int value = *pointer; /* Dereferencing happens here */
            ^~~~~~~~
2 warnings generated.
scan-build: 2 bugs found.
scan-build: Run 'scan-view /tmp/scan-build-2020-10-15-161857-10509-1' to examine bug reports.

> scan-view /tmp/scan-build-2020-10-15-161857-10509-1
Starting scan-view at: http://127.0.0.1:8181
```

(-> point browser to this)



cppcheck

```
> cppcheck nullpointer.c
Checking nullpointer.c ...
nullpointer.c:7:14: error: Null pointer dereference: pointer [nullPointer]
int value = *pointer; /* Dereferencing happens here */
            ^
nullpointer.c:6:17: note: Assignment 'pointer=NULL', assigned value is 0
int * pointer = NULL;
                ^
nullpointer.c:7:14: note: Null pointer dereference
int value = *pointer; /* Dereferencing happens here */
            ^
```

meta-sca

https://github.com/priv-kweihmann/meta-sca

a collection of tools for static analysis, linting and more

meta-sca - a collection of tools

Is an Openembedded/Yocto Project compatible layer:

- collection of multiple tools around source code analysis
- zero impact - all build-time, no code reaches the target FS
- provides a consistent configuration mechanism
- provides a unified output format
- parsers for cmdline and simple webui (static, pre-rendered)

https://github.com/priv-kweihmann/meta-sca
Maintainer: Konrad Weihmann

priv-kweihmann / **meta-sca**

W

Branch: dunfell | Go to file | Add file | Clone

This branch is 587 commits ahead, 601 commits behind master. | Pull request | Compare

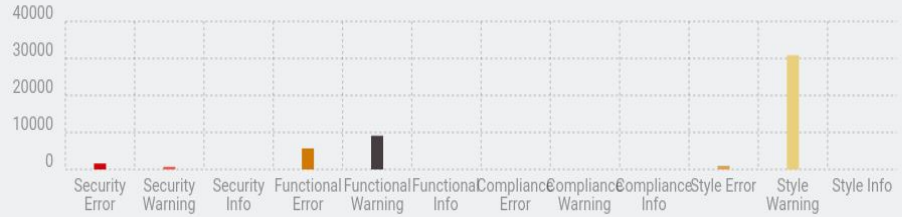
priv-kweihmann committed 998eccb 12 hours ago ... ✓ | **1,873** commits | **12** branches | **62** tags

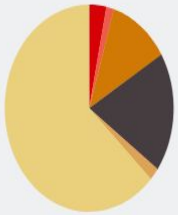
.github/ISSUE_TEMPLATE	Fork dunfell branch	2 months ago
classes	Add shebang checks to pkgqaenc	13 hours ago
conf	npm/composer: stop using in repo tarballs	2 months ago
docs	Add shebang checks to pkgqaenc	13 hours ago
dynamic-distro/scatest	bad-bitbake: set S to source dir	last month
dynamic-layers	Fix php-native install on php 7.4	2 months ago
files	Remove ikos module	19 days ago
recipes-devtools	Update proot-native to latest	12 hours ago

meta-sca status

Here you can see the current findings found by CI pipeline

Get information for branch dunfell and pick a tool all





Show 10 entries

Search:

	PackageName	Tool	Severity	File	ID	Line	Message	Scope
	linux-yocto	bashate	error	.git/hooks/pre-rebase.sample	044	77	Use [[for non-POSIX comparisions	style
	bad-css	stylelint	error	1.css	comment-empty-line-before	8	Expected empty line before comment	style
	bad-css	stylelint	error	1.css	no-missing-end-of-source-newline	17	Unexpected missing end-of-source newline	style
	bad-css	stylelint	error	1.css	no-duplicate-selectors	2	Unexpected duplicate selector "a", first used at line 1	style
	bad-css	stylelint	error	1.css	comment-whitespace-inside	7	Expected whitespace after "/*"	style

Available scanners

Module	Description
alexkohler	Suite of GO analysis tools
ansible	Hardening of images with ansible
ansiblelint	Linter for ansible playbooks
ansibleroles	Hardening of images with 3rd party
bandit	Scan python code for insecurities
bashate	Shell script linter
bitbake	**Bitbake issue handling**
cbmc	C Bounded Model Checker
checkbashisms	Shell script linter
clang	**C/C++ linter using LLVM**
configcheck	Check application configurations
cppcheck	**C/C++ linter**
cpplint	**C/C++ linter**

Available scanners

Module	Description
cspell	Spelling linter
cvecheck	Check for unpatched CVEs
darglint	Python docstring linter
dennis	I18N linter
detectsecrets	Detect hardcoded secrets in code
eslint	Javascript linter
flake8	Python linter
flawfinder	**C/C++ security linter**
flint	**C/C++ linter**
gcc	**GCC compiler issues and hardening**
gixy	NGINX config security linter
golint	GO linter
gosec	GO security linter

Available scanners

Module	Description
govet	GO linter
htmlhint	HTML linter
image-summary	Aggregate all findings in an image
jshint	Javascript linter
jsonlint	JSON file linter
kconfighard	Kernel config hardening checker
looong	Find functions with too long arglists
licensecheck	Scan code for license information
luacheck	LUA linter
lynis	Auditing tool for images
msgcheck	I18n linter
multimetric	Coding metrics
mypy	Python linter

Available scanners

Module	Description
nixauditor	Auditing tool for images
npmaudit	NPM package auditor
oelint	**Bitbake recipe linter**
perl	Perl warnings check
perlcritic	Perl linter
phan	PHP linter
phpcodefixer	Find deprecated PHP functions
phpcodesniffer	PHP, Js and CSS linter
phpmd	PHP Linter
phpsecaudit	Find vulnerabilities in PHP code
phpstan	PHP linter
pkgqaenc	Enhanced package QA
progpilot	PHP linter with security focus

Available scanners

Module	Description
proselint	Spelling and text linter
pscan	Find insecure printfs
pyfindinjection	Find SQL injections in python code
pylint	Python linter
pyright	Python type linter
pysymcheck	Check binaries for forbidden func use
pytype	Python linter using type-annotations
rats	**Check insecurities in several languages**
reconfbf	security audit tool
reek	Code smell detector for Rub
retire	vulnerabilities in javascript and NPM
revive	GO linter
ropgadget	Check ROP exploitability in binaries

Available scanners

Module	Description
rubycritic	Ruby linter
safety	vulnerabilities in python-packages
setuptoolslint	Lint python-setup.py
shellcheck	Shell script linter
slick	Shell script linter
sparse	**C linter**
splint	**C linter**
standard	Javascript linter
stank	Shell script linter
stylelint	CSS/SCSS linter
sudokiller	check on sudo
systemdlint	Systemd unit linter
textlint	Spelling and text linter

Available scanners

Module	Description
tiger	security audit/intrusion detection tool
tlv	Find duplicate code
tscancode	C and lua linter
upc	check for simple privilege escalation
vulture	Find dead python code
wotan	Typescript/javascript linter
xmllint	XML linter
yamllint	YAML linter
yara	Find suspicious pattern in binaries

Phew 87 options … a lot !

Multiple Categories and Scopes

- Language specific scanners
  - C/C++, Python, Perl, PHP, JS, Go, Lua
- Spelling, Metrics
- Scopes:
  - Security
  - Functional
  - Style

Example: step-by-step

```
git clone https://github.com/kraj/meta-clang.git
git clone https://github.com/priv-kweihmann/meta-sca.git

# (check the meta-sca README.md, there is also a conf script)
git clone https://git.yoctoproject.org/git/poky
source poky/oe-init-build-env build-test-sca

bitbake-layers add-layer ../meta-sca
bitbake-layers add-layer ../meta-clang

Next: edit conf/local.conf
```

Example: step-by-step

```
cat << EOF >> conf/local.conf
INHERIT += "sca"
SCA_ENABLE = "1"
#SCA_SPARE_LAYER = "core yocto yoctobsp openembedded-layer"
SCA_AUTO_INH_ON_IMAGE = "1"
SCA_AUTO_INH_ON_RECIPE = "1"
SCA_AUTO_LICENSE_FILTER = ".*"
SCA_AVAILABLE_MODULES = "rats clang cvecheck"
SCA_ENABLED_MODULES_RECIPE = "rats clang cvecheck"

# continues on next slide
```

Example: step-by-step

```
MYSCA_DONOTSCAN += "linux-libc-headers linux-yocto gcc libgcc \\
    gobject-introspection clang compiler-rt boost libcxx "
SCA_BLACKLIST_rats += "\${MYSCA_DONOTSCAN}"
SCA_BLACKLIST_clang += "\${MYSCA_DONOTSCAN}"

# workaround bbappend which fails on non-standard DL_DIR
CVE_CHECK_DB_DIR = "\${DL_DIR}/CVE_CHECK"

EOF

bitbake core-image-minimal
```

Example: step-by-step

```
../meta-sca/scripts/results2console tmp/deploy/images/qemux86-64/sca/ > out
```

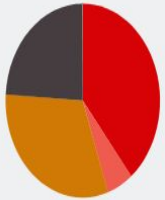
```
grep base-passwd out
```

```
clang@base-passwd: update-passwd.c:399:10 - [warning] - [clang.clang-analyzer-unix.Malloc] - Potential leak of memory pointed to by 'node'

clang@base-passwd: update-passwd.c:475:10 - [warning] - [clang.clang-analyzer-unix.Malloc] - Potential leak of memory pointed to by 'node'

clang@base-passwd: update-passwd.c:438:10 - [warning] - [clang.clang-analyzer-unix.Malloc] - Potential leak of memory pointed to by 'node'

clang@base-passwd: update-passwd.c:158:12 - [warning] - [clang.clang-analyzer-security.insecureAPI.strcpy] - Call to function 'strcpy' is
        insecure as it does not provide bounding of the memory buffer. Replace unbounded copy functions with analogous functions that
        support length arguments such as 'strlcpy'. CWE-119


rats@base-passwd: update-passwd.c:1136:1 - [error] - [rats.rats.getopt_long] - Truncate all input strings to a reasonable length before passing them

rats@base-passwd: update-passwd.c:1212:1 - [error] - [rats.rats.umask] - umask() can easily be used to create files with unsafe priviledges.

rats@base-passwd: update-passwd.c:898:1 - [warning] - [rats.rats.lstat] - A potential TOCTOU (Time Of Check, Time Of Use) vulnerability exists.
        This is the first line where a check has occured. The following line(s) contain uses that may match up with this check: 882 (rename)

rats@base-passwd: update-passwd.c:915:1 - [warning] - [rats.rats.lstat] - A potential TOCTOU (Time Of Check, Time Of Use) vulnerability exists.
        This is the first line where a check has occured. The following line(s) contain uses that may match up with this check: 903 (chmod) , 908 (lchown)

rats@base-passwd: update-passwd.c:831:1 - [error] - [rats.rats.fprintf] - Check to be sure that the non-constant format string passed as argument 2
        to this function call does not come from an untrusted source that could have added formatting characters that the code is not prepared to handle
```

meta-sca status

Here you can see the current findings found by CI pipeline

Get information for branch test and pick a tool all



Show 10 entries

Search: base-passwd

PackageName	Tool	Severity	File	ID	Line	Message	Scope
base-passwd	rats	error	update-passwd.c	getopt_long	1136	Truncate all input strings to a reasonable length before passing them to this function	security
base-passwd	rats	error	update-passwd.c	umask	1212	umask() can easily be used to create files with unsafe priviledges. It should be set to restrictive values.	security
base-passwd	rats	error	update-passwd.c	fprintf	831	Check to be sure that the non-constant format string passed as argument 2 to this function call does not come from an untrusted source that could have added formatting characters that the code is not prepared to handle.	security
base-passwd	clang	warning	update-passwd.c	clang-analyzer-unix.Malloc	399	Potential leak of memory pointed to by 'node'	functional
base-passwd	clang	warning	update-passwd.c	clang-analyzer-unix.Malloc	475	Potential leak of memory pointed to by 'node'	functional
base-passwd	clang	warning	update-passwd.c	clang-analyzer-unix.Malloc	438	Potential leak of memory pointed to by 'node'	functional

Summary, pros and cons

++++++++++++++++++++

- meta-sca can be used to easily instrument builds
- can be used for linting and format-checks in CI
- lots of tools pre-integrated to choose from
- unified report format

- cmdline reporting:
  - needs to be parsed/evaluated/filtered to be useful
- postprocessing required to produce simple webpage

meta-codechecker

https://github.com/dl9pf/meta-codechecker

bitbake integration for

https://github.com/Ericsson/codechecker

meta-codechecker - clang-sa/clang-tidy integrated

https://github.com/Ericsson/codechecker

Collection of tools to

- intercept and log the build calls
- analyse the gathered data using (clang-tidy and clangSA)
- report (static or webui)

Extension and successor of the original clang static analyser / scan-build.

Ericsson / codechecker

Watch 49 | Star 972 | Fork 111

Code | Issues 183 | Pull requests 35 | Actions | Projects 3 | Security | Insights

Branch: master | Go to file | Add file | Code

gyorb committed dbf5618 7 days ago ... | 3,818 commits | 5 branches | 36 tags

.github/ISSUE_TEMPLATE	[GitHub] Fix minor grammatical things in the issue templates	7 months ago
analyzer	[analyzer] Fix analyzer --file option	20 days ago
bin	[license] Change license (**#2729**)	last month
codechecker_common	Add a missing space in a debug warning	last month
config	Adding new checkers to the profiles, setting severities	2 months ago
docker	new dockerfiles for test environments	2 years ago
docs	[tools] tu_collector get dependent source files for headers	21 days ago
requirements_py/docs	Merge pull request **#1935** from gyorb/readthedocs	15 months ago
scripts	[license] Change license (**#2729**)	last month

About

CodeChecker is an analyzer tooling, defect database and viewer extension for the Clang Static Analyzer and Clang Tidy

codechecker.readthedocs.io

clang | cpp | c | clang-tidy | static-analysis | linux | results-viewer | macosx | codechecker | llvm | analysis | database | objective-c | defects | docker | static-analyzer | static-analyzers

Readme

Apache-2.0 License

CodeChecker 6.12 Default

Back to product list

Runs 5 | Checker statistics | All reports | New features | agl-service-gps@oneshot

Search for runs...

Diff Delete

Diff	Name	Number of unresolved reports	Detection status	Analyzer statistics	Storage date	Analysis duration	Check command	Version tag	Description	CodeChecker version	Delete
	agl-service-gps@oneshot	1	(1)	clangsa: ✔ (1) clang-tidy: ✔ (1)	2020-07-02 08:41:01	00:00:01	Show			6.13 (dbf5618c00b26f41197d8fa2f1599a3758909924)	
	cynagora@oneshot	17	(17)	clang-tidy: ✔ (30) clangsa: ✔ (30)	2020-07-02 08:00:16	00:00:35	Show			6.13 (dbf5618c00b26f41197d8fa2f1599a3758909924)	
	app-framework-binder@oneshot	79	(79)	clangsa: ✔ (92) ✖ (3) clang-tidy: ✔ (92) ✖ (3)	2020-07-02 07:50:44	00:02:04	Show			6.13 (dbf5618c00b26f41197d8fa2f1599a3758909924)	
	app-framework-main@oneshot	35	(36)	clangsa: ✔ (34) clang-tidy: ✔ (34)	2020-07-01 22:04:52	00:00:43	Show			6.13 (dbf5618c00b26f41197d8fa2f1599a3758909924)	
	agl-service-audiomixer	4	(4)	clang-tidy: ✔ (2) clangsa: ✔ (2)	2020-07-01 21:36:00	00:00:01	Show			6.13 (dbf5618c00b26f41197d8fa2f1599a3758909924)	

Overview

Userspace tool CodeChecker is a set of python helpers

- main feature is that you wrap you build commands like so

```
CodeChecker log -b "make" -o compilation.json
```

- This will preload a logger and store the compiler commands
- With the exact commands logged, we can replay the compilation using clang and its tools clang-tidy and clangSA

```
CodeChecker analyze compilation.json -o ./reports
```

Overview

- From there you can 'parse' into reports

```
CodeChecker parse ./reports

CodeChecker parse ./reports -e html -o
reports_html
```

- or 'store' online in webui/frontend

```
CodeChecker store ./reports --name mypkg@v0.9 \
    --url http://localhost:8001/Default
```

CodeChecker 6.12 Default

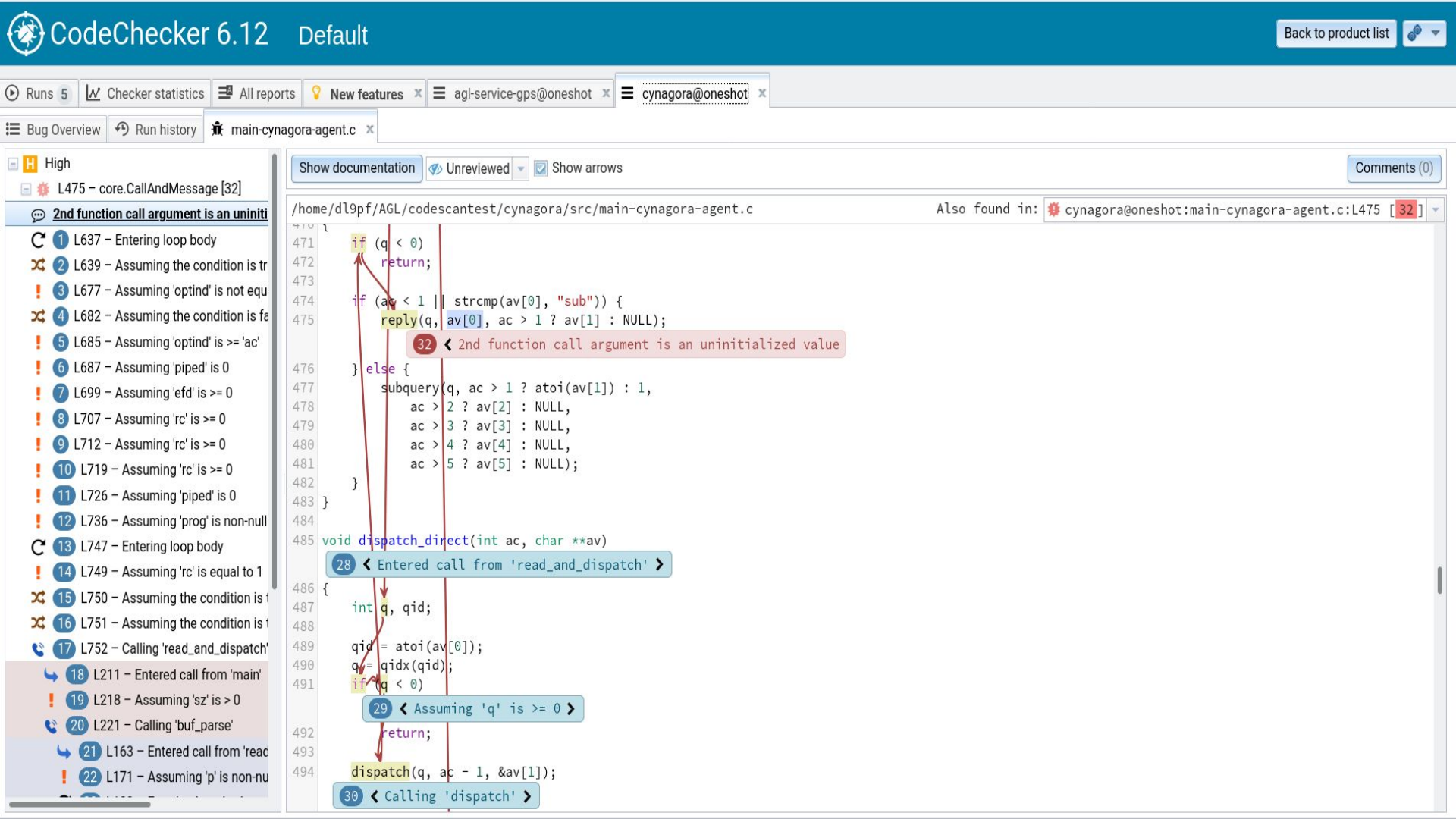
Back to product list

Runs 5 | Checker statistics | All reports | New features

Search for runs...

Diff Delete

Diff	Name	Number of unresolved reports	Detection status	Analyzer statistics	Storage date	Analysis duration	Check command	Version tag	Description	CodeChecker version	Delete
	agl-service-gps@oneshot	1	(1)	clangsa: ✔ (1) clang-tidy: ✔ (1)	2020-07-02 08:41:01	00:00:01	Show			6.13 (dbf5618c00b26f41197d8fa2f1599a3758909924)	
	cynagora@oneshot	17	(17)	clang-tidy: ✔ (30) clangsa: ✔ (30)	2020-07-02 08:00:16	00:00:35	Show			6.13 (dbf5618c00b26f41197d8fa2f1599a3758909924)	
	app-framework-binder@oneshot	79	(79)	clangsa: ✔ (92) ✖ (3) clang-tidy: ✔ (92) ✖ (3)	2020-07-02 07:50:44	00:02:04	Show			6.13 (dbf5618c00b26f41197d8fa2f1599a3758909924)	
	app-framework-main@oneshot	35	(36)	clangsa: ✔ (34) clang-tidy: ✔ (34)	2020-07-01 22:04:52	00:00:43	Show			6.13 (dbf5618c00b26f41197d8fa2f1599a3758909924)	
	agl-service-audiomixer	4	(4)	clang-tidy: ✔ (2) clangsa: ✔ (2)	2020-07-01 21:36:00	00:00:01	Show			6.13 (dbf5618c00b26f41197d8fa2f1599a3758909924)	


CodeChecker 6.12
Default
Back to product list
Runs 5
Checker statistics
All reports
New features
agl-service-gps@oneshot
cynagora@oneshot
Bug Overview
Run history
main-cynagora-agent.c
High
L475 – core.CallAndMessage [32]
2nd function call argument is an uniniti
1 L637 – Entering loop body
2 L639 – Assuming the condition is tr
3 L677 – Assuming 'optind' is not equ
4 L682 – Assuming the condition is fa
5 L685 – Assuming 'optind' is >= 'ac'
6 L687 – Assuming 'piped' is 0
7 L699 – Assuming 'efd' is >= 0
8 L707 – Assuming 'rc' is >= 0
9 L712 – Assuming 'rc' is >= 0
10 L719 – Assuming 'rc' is >= 0
11 L726 – Assuming 'piped' is 0
12 L736 – Assuming 'prog' is non-null
13 L747 – Entering loop body
14 L749 – Assuming 'rc' is equal to 1
15 L750 – Assuming the condition is t
16 L751 – Assuming the condition is t
17 L752 – Calling 'read_and_dispatch'
18 L211 – Entered call from 'main'
19 L218 – Assuming 'sz' is > 0
20 L221 – Calling 'buf_parse'
21 L163 – Entered call from 'read
22 L171 – Assuming 'p' is non-nu
Show documentation
Unreviewed
Show arrows
Comments (0)
/home/dl9pf/AGL/codescantest/cynagora/src/main-cynagora-agent.c
Also found in: cynagora@oneshot:main-cynagora-agent.c:L475 [32]
471 if (q < 0)
472 return;
474 if (ac < 1 || strcmp(av[0], "sub")) {
475 reply(q, av[0], ac > 1 ? av[1] : NULL);
32 2nd function call argument is an uninitialized value
476 } else {
477 subquery(q, ac > 1 ? atoi(av[1]) : 1,
478 ac > 2 ? av[2] : NULL,
479 ac > 3 ? av[3] : NULL,
480 ac > 4 ? av[4] : NULL,
481 ac > 5 ? av[5] : NULL);
482 }
483 }
485 void dispatch_direct(int ac, char **av)
28 Entered call from 'read_and_dispatch'
486 {
487 int q, qid;
489 qid = atoi(av[0]);
490 q = qidx(qid);
491 if (q < 0)
29 Assuming 'q' is >= 0
492 return;
494 dispatch(q, ac - 1, &av[1]);
30 Calling 'dispatch'

meta-codechecker - bitbake integration

- Integrates Codechecker seamlessly with bitbake
  - can write HTML reports
  - and upload to database
  - builds all necessary tools on-the-fly
    - requires meta-clang, meta-oe, meta-python

meta-codechecker - Example: step-by-step

```
git clone https://github.com/kraj/meta-clang.git
git clone https://git.openembedded.org/meta-openembedded
git clone https://github.com/dl9pf/meta-codechecker.git

# (check the meta-codechecker'S README.md)
git clone https://git.yoctoproject.org/git/poky
source poky/oe-init-build-env build-test-codechecker

bitbake-layers add-layer ../meta-clang
bitbake-layers add-layer ../meta-openembedded/meta-oe
bitbake-layers add-layer ../meta-openembedded/meta-python
bitbake-layers add-layer ../meta-codechecker

Next: edit conf/local.conf
```

meta-codechecker - Example: step-by-step

```
cat << EOF >> conf/local.conf
INHERIT += "codechecker"

#enable for all target packages:
CODECHECKER_ENABLED_class-target = "1"

# exempt clang
CODECHECKER_ENABLED_pn-clang = "0"

CODECHECKER_REPORT_HTML = "1"
EOF
```

meta-codechecker - Example: step-by-step

```
bitbake core-image-minimal
tree tmp/deploy/CodeChecker/
```

Summary, pros and cons

++++++++++++++++++++

- CodeChecker can be used by developers and in CI
- complexity hidden by pre-loaded logger library
- straightforward workflow
- parsers into multiple formats
- Webui to store and browse/review results
- bitbake integration using meta-codechecker

- documentation is good, but has a few dead links and such

Summary & lookout

- Static Analysis can help improve your projects!
- Easy to use locally for development
- Integration to OpenEmbedded / Yocto Project

- Next:
  - promote use of tools
  - enhance meta-codechecker

Q/A

Questions ?

Answers !

End

Thank you.